Learning Trajectory Dependencies for Human Motion Prediction —Supplementary Material—

Wei Mao[1], Miaomiao Liu[1,3], Mathieu Salzmann[2], Hongdong Li[1,3]
[1]Australian National University, [2]CVLab, EPFL,[3]Australia Centre for Robotic Vision
{wei.mao, miaomiao.liu, hongdong.li}@anu.edu.au, mathieu.salzmann@epfl.ch

1. Datasets

Below, we provide more detail on the datasets used in our experiments.

Human3.6M. In H3.6M, each pose has 32 joints. Removing the global rotation, translation and constant angles, leaves us with a 48 dimensional vector for each human motion, denoting the exponential map representation of the joint angles. Furthermore, a 3D human pose can also be represented by a 66 dimensional vector of 3D coordinates after removing the global rotation, translation and stationary joints across time. We use the same training and test split as previous work [4, 3, 1]. That is, we test our model on the same image sequence of subject 5 as previous work [4, 3, 1]. For training, we keep subject 11 as validation set to choose the best model (the one that achieves the least average error across all future frames) and use the remaining 5 subjects as training set.

CMU-Mocap. In this dataset, we use a 64 dimensional vector to represent every human pose by removing the global rotation, translation and joint angles with constant values. Each component of the vector denotes the exponential map representation of the joint angle. We further use 75 dimensional vectors for the 3D joint coordinate representation. We do not use a validation set due to limited training data.

3DPW. The human skeleton in this dataset uses 24 joints, yielding a 72 dimensional vector for the angle representation. For the 3D joint coordinate one, we obtain a 69 dimensional vector after removing the global translation.

2. Visualizing the Results on H3.6M in Video

We provide more visualization of the results on H3.6M in a video (See the supplementary video). In particular, the video compares our approach with the state of the art on periodic actions, such as walking, and aperiodic ones, such as eating and direction. Our approach shows better performance than the state-of-the-art ones.

Furthermore, in the video, we provide additional (quantitative and qualitative) visualization of cases where large errors are observed according to the angle representation but small errors in 3D space. This confirms that ambiguities exist in angle space for human motion prediction.

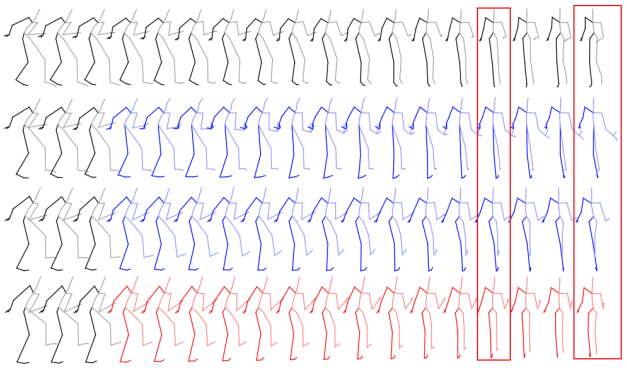

Figure 1. Motion prediction in 3D space on the "basketball action of CMU-Mocap. From top to bottom: Ground truth, results of [4], results of [3] and our results. The highlighted results in the box show that we can make better predictions on the legs and arms of the subject.

3. Visualizing the Results on CMU-Mocap

We provide a qualitative visualisation of the 3D human pose prediction on the "basketball", "basketball signal" and "direction traffic" actions of the CMU-Mocap dataset in Fig. 1, Fig. 2 and Fig. 3, respectively. Again, our approach outperforms the state-of-the-art ones (see highlighted poses).

4. Number of DCT Coefficients

In this section, we first present the intuition behind using fewer DCT coefficients to represent the whole sequence. We then compare the performance of using different number of DCT coefficients.

4.1. Using Fewer Coefficients

Given a smooth trajectory, it is possible to discard some high frequency DCT coefficients without losing prediction

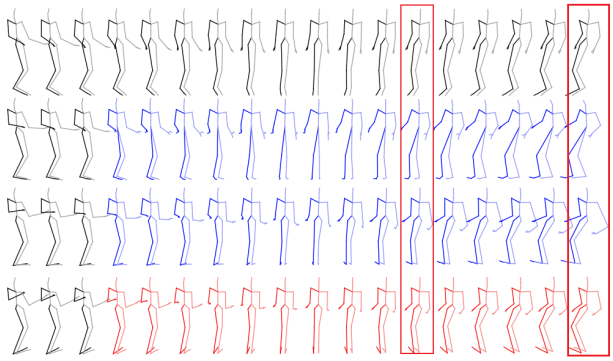

Figure 2. Motion prediction in 3D space on the "basketball signal" action of CMU-Mocap. From top to bottom: Ground truth, results of [4], results of [3] and our results.

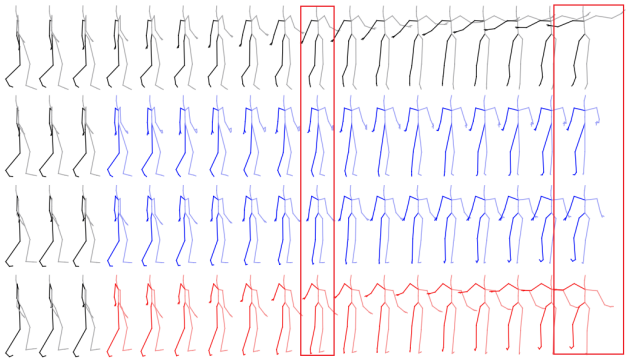

Figure 3. Motion prediction in 3D space on the "directing traffic" action of CMU-Mocap. From top to bottom: Ground truth, results of [4], results of [3] and our results.

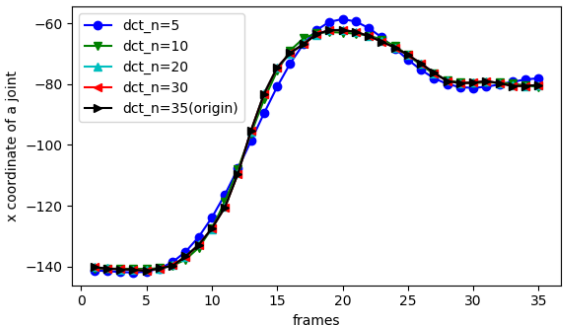

Figure 4. Temporal trajectory of the x coordinate of one joint reconstructed using different number of DCT coefficients.

accuracy. To evidence this, in Fig. 4, we show the effect of the number of DCT components in reconstructing a sequence of 35 frames for the one human joint predicted using our approach. Note that, since we use 35 frames, 35 DCT coefficients yield a lossless reconstruction. Nevertheless, even 10 DCT coefficients are enough to reconstruct the trajectory with a very low error. This is due to the smoothness of the joint trajectory in 3D space.

4.2. Results on H3.6M

Experiment setup. Based on the previous discussion, we perform more experiments to evaluate the influence of the number of input DCT coefficients on human motion prediction. In the following experiments, we assume that we observe 10 frames to predict the future 25 frames. Following the same formulation as in our submission, the observed sequence is padded with the last observed frame replicated 25 times and then transformed to DCT coefficients. The target is the DCT coefficients of the whole sequence (35 frames). We perform several experiments by preserving different number of DCT coefficients. For instance, 'dct_n=5' means that we only use the first 5 DCT coefficients for temporal reconstruction. The experiments are performed on both 3D and angle representation.

Fig. 6 shows the error for short-term prediction at 160ms and long-term prediction at 560ms in angle representation as a function of the number of DCT coefficients. In general, the angle error decreases with the increase of number of DCT coefficients. Similarly, in Fig. 7, we plot the motion prediction error in 3D coordinates at 160ms and 560ms as a function of the number of DCT coefficients. Here, 10 DCT coefficients already give a very small prediction error. Interestingly, when we use more DCT coefficients, the average error sometimes increases (see the plot for prediction at 560ms). This pattern confirms our argument in the submission that the use of truncated DCT coefficients can prevent a model from generating jittery motion, because the 3D coordinate representation of human motion trajectory is smooth.

To analyse the different patterns of the prediction error w.r.t. the number of DCT coefficients shown in angle representation (Fig. 6) and 3D representation (Fig. 7), we looked into the dataset and found that there are large discontinuities in the trajectories of angles. As shown in Fig. 8, these large jumps make the reconstruction of trajectories with fewer DCT coefficients lossy.

In summary, we can discard some of the high frequency coefficients to achieve better performance in 3D space. In our experiments, we use the first 15 DCT coefficients as input to our network for short-term prediction and 30 coefficients for long-term prediction in 3D space. As the joint trajectory in angle representation is not smooth and has large discontinuities, we therefore take the full frequency as input to our network for motion prediction in angle representation. In our experiments, we therefore use 20 DCT coefficients as input to our network for short-term prediction and 35 for long-term prediction in angle representation.

5. Ablation Study Details

Fully-connected Network. In our ablation study, we also compare the motion prediction using a graph network

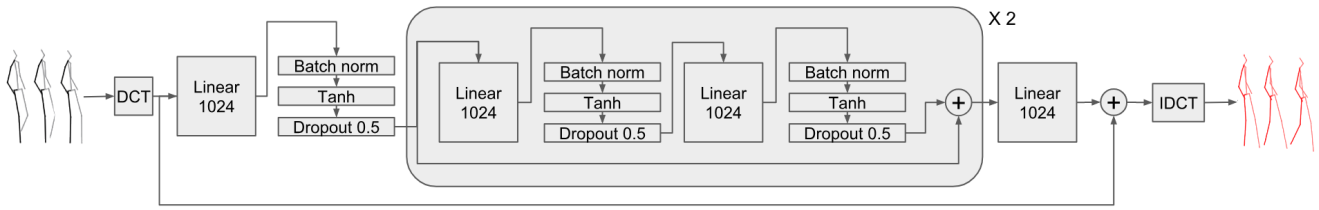

Figure 5. Fully-connected Network Structure

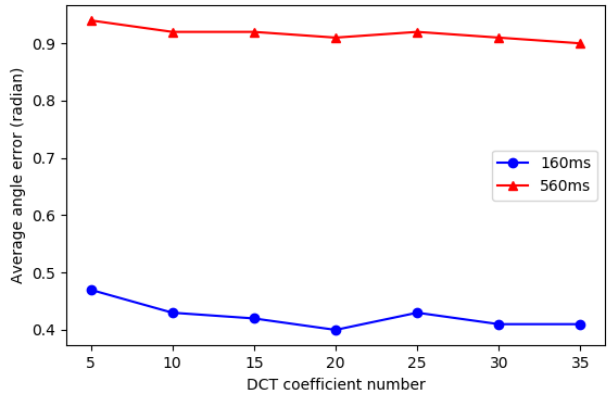

Figure 6. Average angle prediction error over 4 actions (”walking”,”eating”,”smoking”,”discussion”) using different number of DCT coefficients at 160ms (blue) and 560ms (red).

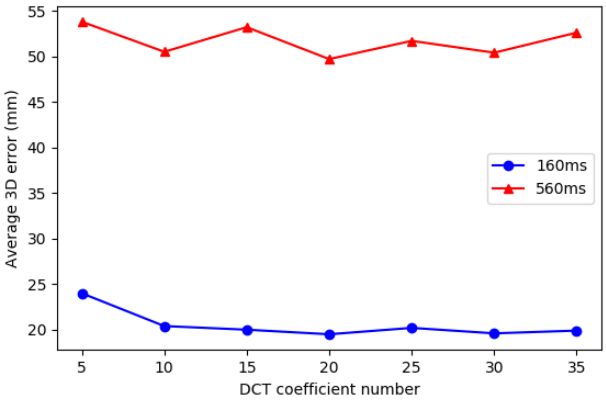

Figure 7. Average 3D prediction error over 4 actions (“walking”,“eating”,“smoking”,“discussion”) using different number of DCT coefficients at 160ms (blue) and 560ms (red).

with that of a fully-connected network structure. We apply the same process of encoding temporal information via the DCT. Before being fed to the network, the DCT coefficients of the past sequence padded with last frame are flattened to a vector and the network learns the residual between the past temporal encoding and the future one. To this end, we adopt the network structure shown in Fig. 5. Instead of using graph convolutional layers, we rely on 2 fully connected layers with residual connections. We additionally use two fully connected layers at the start of the network for encoding the DCT coefficients and at the end for decoding the feature to the residual of the DCT coefficient.

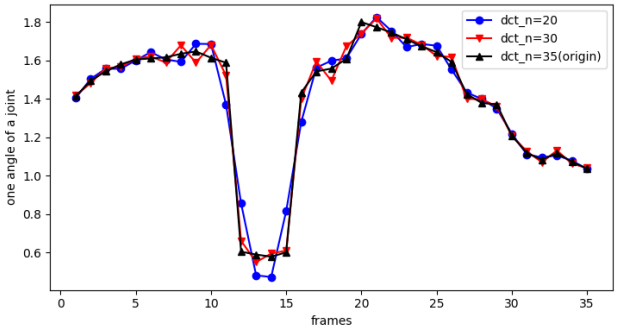

Figure 8. The temporal trajectory of one joint angle reconstructed using different number of DCT coefficients. Note that the trajectory is not smooth and has large jumps. The full frequency (35 DCT coefficients) leads to lossless temporal reconstruction of the trajectory.

The implementation details for this network are the same as our Graph Convolutional Network. We implemented this network using Pytorch [5], and we used ADAM [2] to train this model. The learning rate was set to 0.0005 with a 0.96 decay every two epochs. The batch size was set to 16 and the gradients were clipped to a maximum $\ell 2$-norm of 1. The model was trained for 50 epochs. As reported in the submission, the fully-connected network structure cannot learn a better representation than the Graph Convolutional Network.

6. Mean Pose Problem

As explained in [3], the mean pose problem typically occurs when using recurrent neural networks (RNNs) to encode temporal dynamics, where the past information may vanish during long propagation paths. By not relying on RNNs, but directly encoding the trajectory of the whole sequence, our method inherently prevents us from losing the past information. This is evidenced by Fig. 9, where our method yields poses significantly further from the mean pose than the RNN-based method [4].

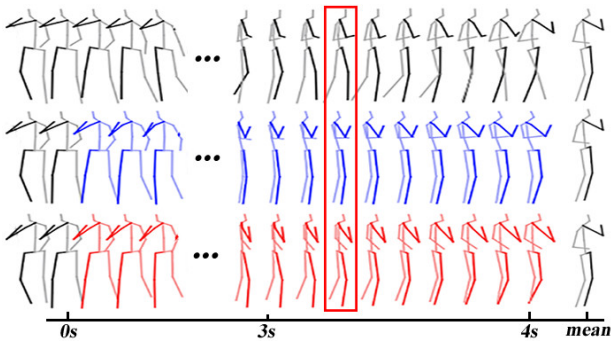

Figure 9. Prediction up to 4 seconds for the Phoning action of Human3.6m. From top to bottom, we show the ground truth, the poses predicted by [4] , and by our method. Note that, after the highlighted frame, the poses predicted by the RNN of [4] have indeed converged to the mean pose (shown in the last column), whereas in our predictions the legs continue to move.

References

[1] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *ECCV*, pages 786–803, 2018. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[3] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *CVPR*, pages 5226–5234, 2018. 1, 2, 3

[4] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, July 2017. 1, 2, 3, 4

[5] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 3